Writing Good Code

A short primer.
Consistency is Key

Style Guides are good
(Hadley, DuckDuckGo)
Naming Things

Good names are the first step to good code.

• underscore_separated

• camelCase

• dot.separated

• Be consistent with use of plurals.

Aim to be concise and meaningful

variables = nouns
functions = verbs
Good Indentation

(and keep line lengths short)

- Essential for readable code
- RStudio will do it for you!
- BE CONSISTENT
- Code within curly braces ( { } ) should always be indented
  (and opening curly braces should not be on a line on their own)
Whitespace is your Friend!

- Place spaces around all infix operators (=, +, -, <-, etc.).

- Always put a space after a comma, and never before (just like in regular English).

- Place a space before left parentheses, except in a function call.

- Use empty lines to separate cognitive pieces of code within functions.
Comments

• At minimum, before each function definition should be a comment explaining what it does and its inputs and outputs.

• Anything unclear should also get a clarifying comment.

• Cleverer / more concise code often requires better comments.
Functions and Breaking Code Up

• Good code is broken up into functions.
• Each function should do one well defined thing
• Names of functions should tell you what they do
• Don’t Copy Code (don’t repeat yourself, DRY)
Performance

- Especially at first, worry about your code being clear and working first.
- For loops aren’t the fastest, but are easy to read and understand as you are learning.
Writing Good Code is like writing good prose

• Code is a means of communication and should be written as such.

• You’re always writing for another person!

• Journals are often requiring you to include your code, put as much work into the language of your code as the language of your paper.